基于图神经网络的工业过程控制回路故障诊断

孟德森[1]
[1]（北京化工大学信息科学与技术学院）

摘要：
本文基于图神经网络，提出了一种用于工业过程控制回路故障诊断的方法。通过对回路传感器输出信号的监测，图神经网络能够捕捉到回路中的异常行为，并自动诊断回路故障类型。实验结果表明，该方法能够高效地检测到回路故障，并且能够在单故障和多故障情况下都实现较高的准确率。该方法为工业过程控制提供了可靠的故障诊断方案，在实际工业应用中具有重要的意义和应用价值。
关键词：图神经卷积神经网络　控制回路　故障诊断

FAULT DIAGNOSES OF INDUSTRIAL PROCESS CONTROL LOOPS BASED ON GRAPH NEURAL NETWORKS

Desen Meng[1]
[1]College of Information Science and Technology, Beijing University of Chemical Technology

Abstract：
This thesis proposes a method for industrial process control loop fault diagnosis based on graph neural networks. By monitoring the output signals of loop sensors, the graph neural network can capture abnormal behaviors in the loop and automatically diagnose the type of loop faults. Experimental results demonstrate that the proposed method can efficiently detect loop faults and achieve high accuracy in both single and multiple fault scenarios. This method provides a reliable fault diagnosis solution for industrial process control, which has important practical significance and application value in actual industrial applications.
KEY WORDS: Graph Convolutional Neural Networks Control Loop Fault Diagnosis

1 绪论

随着科技和社会的发展，工业过程和控制系统日益复杂，自动控制技术广泛应用。控制回路是现代工业控制系统的基础，由控制器、控制阀和变送器组成，用于稳定参数指标。然而，在实际工业过程中，控制回路可能发生故障，导致产品质量下降和成本增加。为了确保工业生产的稳定和安全，故障诊断至关重要。

传统故障诊断方法依赖专家知识和经验，效率低且精度不高。数据驱动的方法，尤其是基于神经网络的方法，近年来备受关注。然而，传统神经网络在处理非欧几里德结构数据方面存在限制。为了克服这些问题，图神经网络(GNN)成为研究热点，通过引入图结构和考虑数据之间的相关性，提高了故障诊断的精度和效率[1]。

在工业控制回路中，故障点与相邻测量数据具有一定的相关性。基于这一观察，将图神经网络应用于控制回路的故障诊断能够更好地完成相关工作，并提供对数据的解释支持。因此，本研究旨在验证图神经网络在工业控制回路故障诊断中的

性能是否优于传统的卷积神经网络模型。值得注意的是，图神经网络的可解释性是其在故障诊断领域的优势之一[2]。

2 基础理论

2.1 图卷积神经网络(GCN)原理

图卷积神经网络（GCN）是一种用于处理图结构数据的深度学习算法。GCN 的主要思想是在图上进行卷积操作，类似于在图像上进行卷积操作。在 GCN 中，节点和边都被视为网络的输入数据。GCN 使用邻接矩阵和节点特征矩阵来进行卷积操作，邻接矩阵表示节点之间的连接关系，节点特征矩阵表示节点的特征向量。

GCN 的卷积过程可以分为两步。首先，通过邻接矩阵来计算每个节点与其邻居节点的关系，这个过程被称为消息传递。其次，将节点的特征向量和消息向量进行融合，得到卷积后的节点特征向量。这个过程可以通过对节点特征矩阵和邻接矩阵进行乘法运算和加法运算实现，即每个节点都能提取邻接节点的特征。

GCN 的层数可以根据需要增加，每一层的输出作为下一层的输入。通过堆叠多个 GCN 层，可以提高模型的深度和复杂度，从而更好地捕捉图结构中的信息和关系。但是由于 GCN 的特殊性——在图结构上进行卷积操作，这大大减少了 GCN 所需要的层数。举例说明，节点 A1 在第一层都获得了邻接节点 B 的特征值，输出 A2，而节点 B1 在第一层获得了邻接节点 C、D 的特征值，输出 B2。那么在第二层中，节点 A2 再提取邻接节点 B2 的特征值，其实也提取到了 C、D 的特征值。这个特点可以让 GCN 用很少的层数获取全局信息[3]。

总的来说，GCN 是一种用于处理图结构数据的深度学习算法，它通过在图上进行卷积操作来提取节点的特征，可以实现对图数据的分类、预测和表示学习等任务。

2.2 控制回路 GCN 结构

控制回路 GCN 结构中的关键环节是构建图结构。图神经网络中的图结构数据是由节点特征和节点间的边关系组成的，在控制回路中，选择控制器输出（OP）、控制阀输出（MV）、回路输出（PV）和变送器输出（TO）四组数据作为控制回路图结构的节点特征。对于边关系的选择，有无向图和有向图两个选择，有向图和无向图都可以用于表示变量之间的耦合关系，但是它们各自的特点不同[4]。

无向图通常用于描述物理系统、社交网络、交通网络等领域。这些领域中，变量之间的相互影响和对称性往往更为重要，因此无向图能够更好地表示变量之间的关系。此外，无向图的邻接矩阵通常是对称矩阵，使得在构建 GCN 时更为简便。

有向图通常用于描述生物网络、电路网络等领域。这些领域中，变量之间的因果关系和流向性更为重要，因此有向图能够更好地表示变量之间的关系。有向图的邻接矩阵通常是非对称矩阵，需要更为复杂的计算方式[5]。

在工业过程控制回路中进行故障诊断时，考虑到变量之间的复杂耦合关系以及因果关系和流向性，选择使用有向图来构建控制回路的图结构更为合适。具体而言：

①控制器输出与控制阀输出之间的边关系：控制器输出作为控制阀的输入信号，因此在有向图结构中，应该从控制器输出节点指向控制阀输出节点。

②控制阀输出与控制回路输出之间的边：控制阀输出经过一定的处理后，会得到控制回路输出信号，因此在有向图结构中，应该从控制阀输出节点指向控制回路输出节点。

③控制回路输出与变送器输出之间的边：控制回路输出作为变送器的输入信号，因此在有向图结构中，应该从控制回路输出节点指向变送器输出节点。

④因为有反馈回路，则需要从变送器输出节点指向控制器，表示反馈回路中的信

号作为控制器输入的一部分[6,7]。

综上，构建出的控制回路图结构如图 2-1 和 2-2 所示。

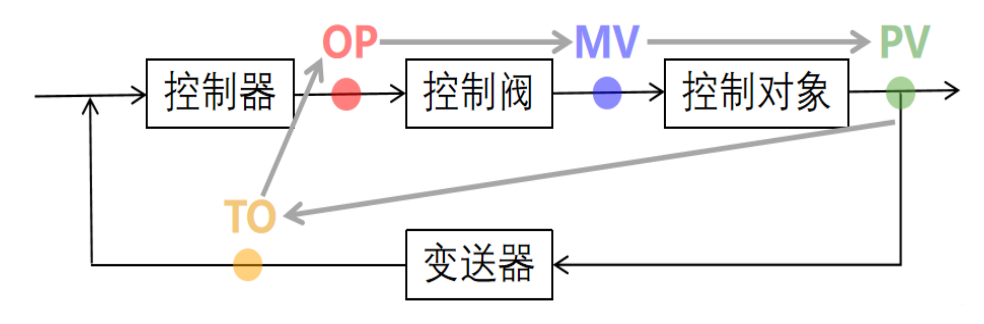


图 2-1 控制回路结构

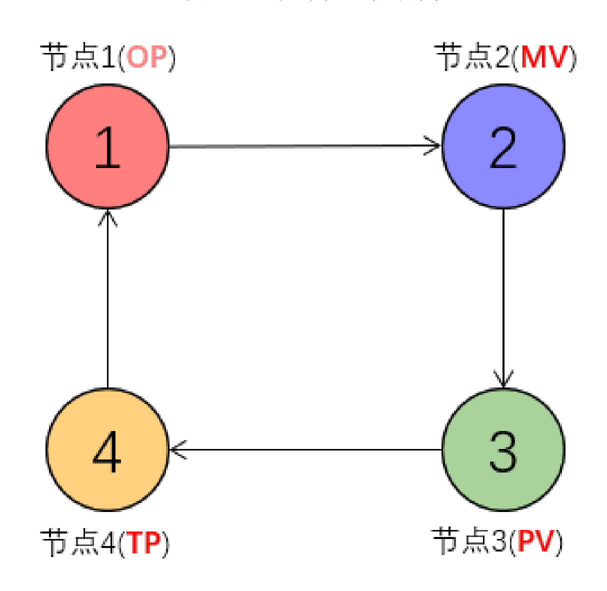


图 2-2 控制回路图结构

确定了节点特征和邻接矩阵，就可以构建出图数据集，分为训练集和测试集。将训练集放进图卷积神经网络的分类模型中进行模型训练。步骤如下：

定义一个基于 GCN 的分类模型，包含 2-3 个 GCN 层和激活函数、正则化操作等，定义优化器和损失函数。在 GCN 分类模型中循环训练批量数据，每次循环都要经过梯度清零、前向传播、计算损失值、反向传播和参数优化，经过多次迭代训练后，GCN 模型的性能会逐步提升。

模型训练完成后，将测试集输入 GCN 模型对测试数据进行预测，然后计算预测的准确率。步骤如下：将测试数据集分批打乱，输入训练好的 GCN 模型。在测试过程中只需要进行前向传播，得到预测结果。将预测结果对应的最大概率类别作为预测的类别，与真实类别标签进行对比，计算并输出最后的预测准确率[8]。

对控制回路 GCN 的原理进行总结：

通过向 GCN 模型中输入图结构形式的训练集和测试集，并通过前向传播将节点和边的特征信息进行整合，输出每个节点的特征表示。接着，将这些特征表示输入全连接层进行分类，模型可以对图数据进行分类任务。训练过程中，模型会使用训练数据对参数进行更新所，通过多次迭代训练，不断优化模型的性能。在测试过程中，将测试数据输入到已经训练好的模型中，进行前向传播，得到每个节点的预测结果，通过取预测概率最大的类别作为预测结果。通过这种方法，可以有效地对图数据进行分类任务。

2.3　学习算法

在图神经网络中，用邻接矩阵 A、度矩阵 D 来表示节点间的关系。

邻接矩阵 A 用来表示图结构中节点之间的关系，在有向图中，如果节点 N_1 和 N_2 之间有 N_1 指向 N_2 的边，则 E_{12}=1，否则=0。至于无向图，如果 N_1 和 N_2 之间有边连接，则 E_{12}=E_{21}=1。

度矩阵 D 中，度表示每个节点与其相连接边的数目，各个节点的度在矩阵对角线上，组成的矩阵就是度矩阵。

邻接矩阵和度矩阵的例图如图 2-3 所示。

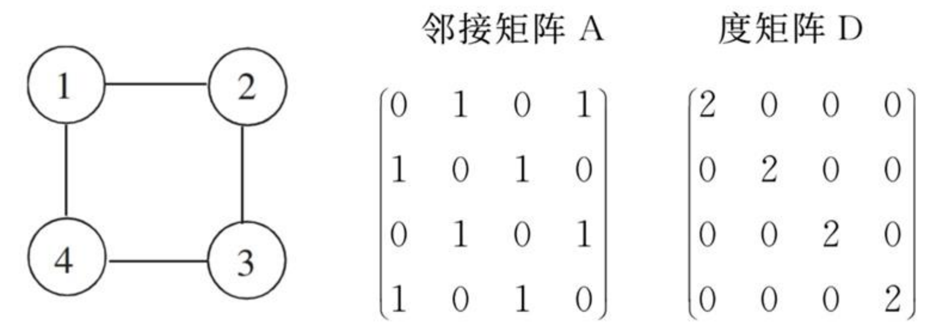


图 2-3 邻接矩阵和度矩阵例图

一组图数据由 N 个节点组成，每个节点有 M 个特征值，可以得到一个大小为 N×M 的卷积特征矩阵 X，和 N×N 的邻接矩阵 A，A 和 X 即为 GCN 模型的输入。

GCN 网络提取目标节点邻接点特征信息，通过一般 2-3 个传播层来获得全局信息，更新目标节点特征值。前向传播的图卷积神经网络，每一层的输出 Hl+1 为一个非线性函数表示，可描述为式 2-1。

$$H^{l+1} = \sigma\left(\hat{D}^{\frac{-1}{2}}\hat{A}\hat{D}^{\frac{-1}{2}}H^{l}W^{l}\right) \qquad (2\text{-}1)$$

其中，Hl 是当前层输入的特征，如果是输入层，则是 $H^0 = X$；$\hat{A}$=A+I，A 是节点的邻接矩阵，I 是单位矩阵；$\hat{D}$是$\hat{A}$的对角线度矩阵；σ是非线性激活函数；Wl 是当前层的权重矩阵[9]。

对于非线性激活函数σ，图神经网络中的每个节点都会接收来自前一层节点的输入，进行一系列线性变换，并通过激活函数进行非线性转换，最终输出给下一层的节点。激活函数的作用是将节点的输出映射到非线性空间，这样图神经网络才能够捕捉到复杂的非线性关系，使图神经网络能够学习到非线性模式。

如果没有激活函数，图神经网络就只能进行线性变换，无法学习到复杂的非线性模式，因此激活函数对于图神经网络的性能至关重要。常用的激活函数有 sigmoid 函数、tanh 函数、ReLU 函数等，在控制回路 GCN 中，选择的是 ReLU 激活函数[10]。

ReLU (Rectified Linear Unit)激活函数是一种常用的非线性激活函数。其表达式为:f(x)=max(0,x)，即当输入小于 0 时，输出为 0，当输入大于等于 0 时，输出等于输入。简单来说，ReLU 激活函数会将小于 0 的值设为 0，大于等于 0 的值不变。

ReLU 激活函数具有简单的计算方式，能够有效地缓解梯度消失问题，并且具有良好的表达能力和加速收敛的效果，因此在深度学习中被广泛使用。

经过全部卷积层，完成数据的前向传播后，目标节点聚合了邻节点高度抽象化的特征，通过进行 1D 全局最大池化操作，得到一个具有全局最大节点特征的输出结果。1D 全局最大池化操作会在每个通道上找到输入序列中的最大值，并将其作为输出张量

对应通道的值。在整个序列中只保留每个通道的最大值，其它位置的值被忽略，这样可以有效地提取出每个通道中的最显著特征，即分类预测的结果。

得到预测结果后，用交叉熵损失函数来判断模型的准确率，描述为式 2-2。

$$L_{CE}(y,\bar{y}) = -\log\left(\frac{e^{y_{\bar{y}}}}{\sum_{j=1}^{C} e^{y_j}}\right) \tag{2-2}$$

其中，y 代表对节点的预测类别，$\bar{y}$ 代表节点的真实类别，C 是节点分类的类数。

得到了模型的损失值 Loss 后，进行反向传播，计算每个参数的梯度，通过优化器更新参数，使当前输入数据上的损失率得到降低。在控制回路 GCN 中，选择的是 Adam 优化器。

Adam (Adaptive Moment Estimation)算法是一种自适应学习率的优化算法，可以用于更新神经网络中的权重参数。它是一种基于梯度下降的算法，结合了 RMSprop 和动量(momentum)方法的思想。

Adam 算法的核心原理是使用动量梯度下降法来更新权重参数。它在计算梯度时，不仅考虑了梯度的一阶矩(平均值)，还考虑了梯度的二阶矩(方差)，并且对二阶矩做了一个指数加权平均。这样做的好处是可以让学习率根据梯度的大小和稳定性来自适应地调整，从而在更新权重参数时更加准确、稳定。其中 Adam 优化器的学习率一般设置在 0.001 至 0.1 之间，学习率设置得较大，可以加快收敛速度，但有可能使模型失去精度；反之，如果学习率设置得较小，则可能需要更多的时间来收敛，但通常可以得到更精确的结果。具体数值需要根据具体问题进行调整。

具体算法的计算流程如下:

①计算梯度:使用反向传播算法计算当前权重参数的梯度。

②计算一阶矩(平均值):对梯度的一阶矩(平均值)进行指数加权平均，得到梯度的移动平均值。

③计算二阶矩(方差):对梯度的二阶矩(方差)进行指数加权平均，得到梯度平方的移动平均值。

④计算偏差校正:由于指数加权平均会使得一阶矩和二阶矩的初值很小，因此需要进行偏差校正，使得平均值更接近真实值。

⑤更新权重参数:根据一阶矩和二阶矩计算出权重参数的更新值，使用动量法更新权重参数。

Adam 算法的优点是具有较快的收敛速度和较好的适应性，可以自适应地调整学习率，同时还具有较强的鲁棒性，不容易受到超参数的影响[11]。

3 实验

3.1 控制回路单故障分类

（1）故障数据集

对于控制回路单故障，通过 Simulink 仿真获取了 45 组正常数据、70 组控制器噪声数据、70 组控制器参数改变数据 、45 组阀门粘滞数据、50 组阀门恒偏差数据、70 组变送器恒增益数据、70 组变送器恒偏差数据，一共 420 组，每组 4×150 个数据。给上述 7 类数据打上 0、1、2、3、4、5、6 的标签，用于构建图标签数据。

首先对数据进行预处理，因为噪声故障、恒偏差故障都在 55s-65s 间加入回路，观察各类故障的数据图像，可以发现 60s-85s 的数据曲线有明显的故障特征。所以在 150s 数据中，只需要提取 56s-95s 的 40 个包含了故障特征的数据作为数据集，大大减少了数据量。

将 420 组 4×40 的控制回路故障数据、预处理好的图结构 4×4 邻接矩阵和每类

故障 1×7 的标签数据都转换成 Tensor 类型，生成控制回路单故障的数据集。

对数据进行处理，每批次数据的数量为 30。此时得到的控制回路单故障数据集包含 14 个 batch，每个 batch 由标签为 x 的 120×40 节点特征集合、标签为 edge_index 的 2×120 边关系集合、标签为 y 的 30×7 的图分类标签集合组成。

流程图如图 3-1 所示。

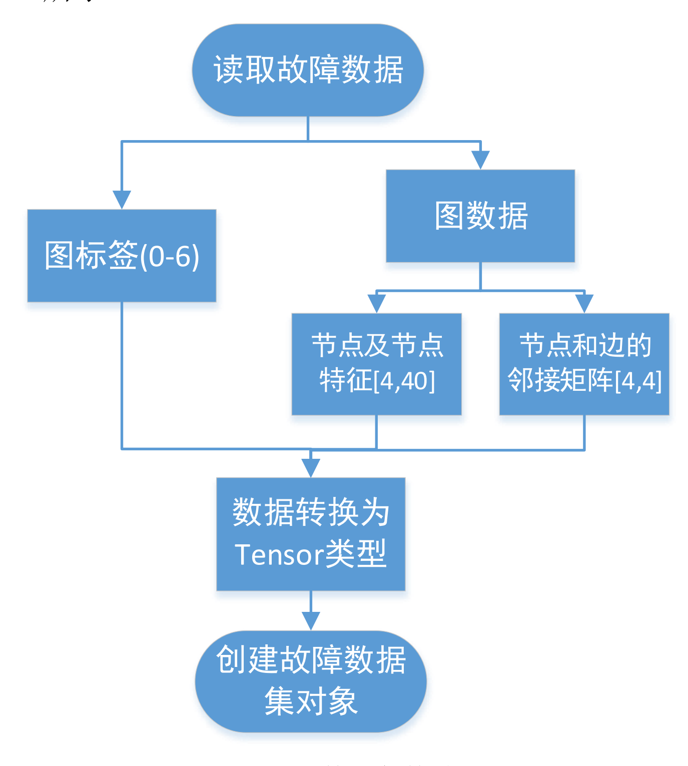


图 3-1 数据集构建

（2）GCN 故障分类

把单故障数据集分为用于训练 GCN 模型性能的训练集和用于测试故障分类模型结果的测试集，将前 360 组数据划分为训练集、后 60 组数据划分为测试集。

在模型训练部分中，首先导入相关的库文件，在搭建好图神经网络模型后，输入训练集数据图，通过向优化器反馈学习速率，在迭代中改进参数，以完成模型的训练。训练具体步骤如下：

①定义了一个单故障分类 GCN 模型，包含两层卷积层和一个全局池化层。

②定义了前向传播函数 forward，其中通过 data 参数输入节点特征和边的稀疏矩阵，并通过 GCN 卷积层和 ReLU 激活函数进行运算，通过两个 GCN 卷积层和池化层输出一个 7 维度结果，对应 7 分类任务的预测结果。

③完成 GCN 模型的构建，并定义了一个 Adam 优化器和一个交叉熵损失函数，其中 Adam 优化器的学习参数设定为 0.005。

④加载训练数据集，并使用 DataLoader 分批次加载每 30 个数据。

⑤开始模型的训练，对所有训练数据集进行 200 次迭代(epoch=200)。

⑥对于每个迭代过程中的每个批次数据，执行以下操作：

梯度清零：保证每次迭代都从零开始计算梯度，防止梯度累积导致训练不稳定，确保参数更新的正确性，同时提高内存效率。

进行前向传播：输入的特征 X 是一个 120×40 的张量，通过两个卷积层的计算，X 降为 120×7。再经过池化层的最大池化操作，X 降为 30×7，即 bacth 为 30 的七分类预测结果。

计算损失函数：用交叉熵损失函数来判断模型的准确率（式 3-2）。

反向传播：从输出层开始，根据链式法则计算每个节点对应的梯度。将梯度沿着网络的反方向传播回输入层。在传播过程中，使用链式法则将上一层的梯度乘以当前层的局部梯度，以计算当前层的梯度。这样，每个参数都可以得到相应的梯度。

更新模型参数：根据计算得到的梯度，使用 Adam 算法来更新模型的参数。用于优化模型、收敛到最优解、适应新数据和提升模型鲁棒性。

记录损失总值：监控训练进度，判断模型收敛，并支持可视化分析。

⑦训练完成后，保存到一个 model 文件，用于后续的测试操作。

流程图如图 3-2 所示。

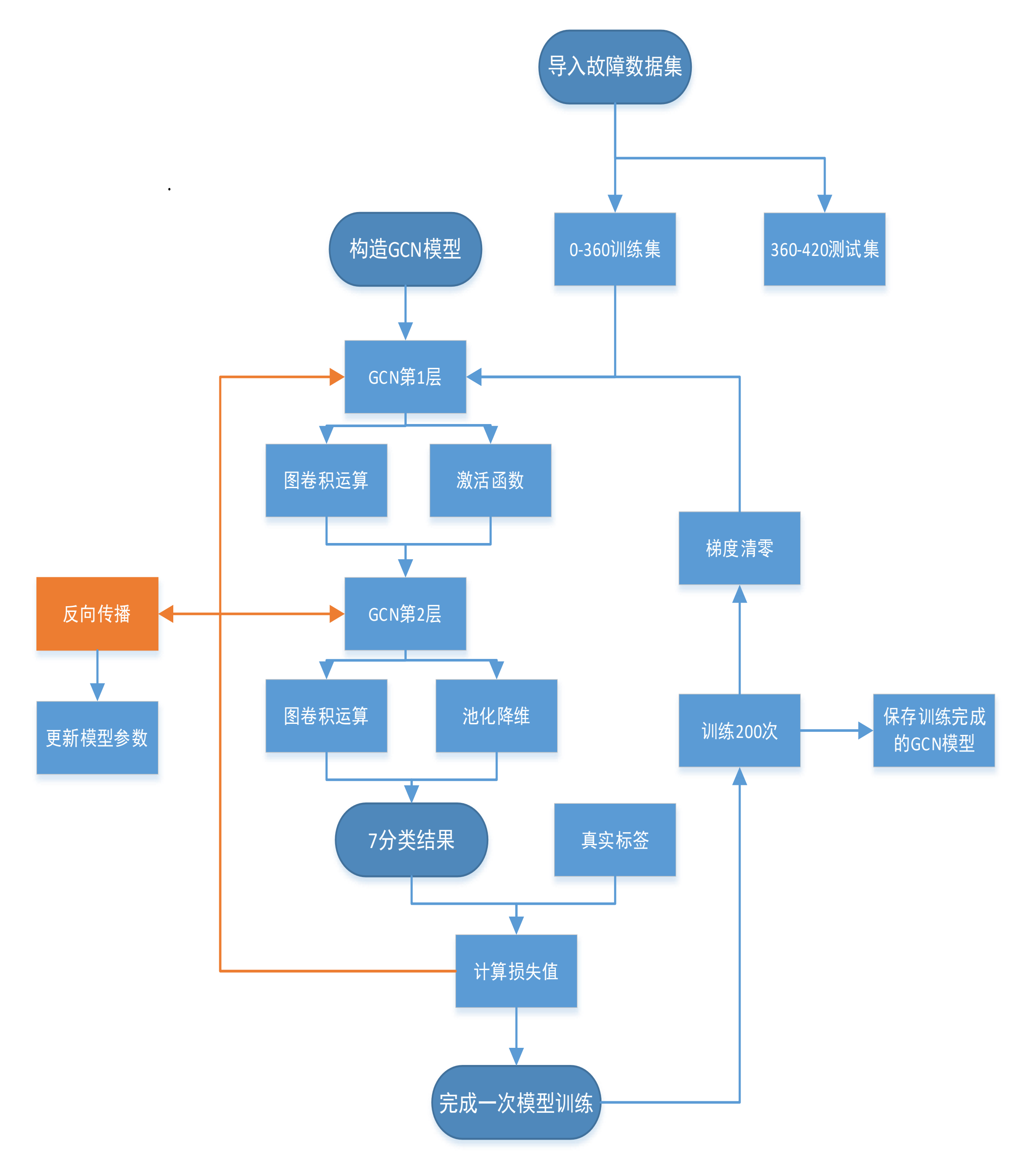


图 3-2 训练 GCN 模型

向训练好的模型输入测试集数据图以得到设计的 GCN 模型对测试集预测的准确率，从而判断模型性能。具体测试步骤如下：

①定义空列表 preds 和 label 来存储预测标签和真实标签。
②使用 DataLoader 函数读取测试数据集，batch=30，打乱数据集
③关闭梯度计算，以提高预测效率。
④对于 loaders 中的每个 batch 中的数据进行预测，其中 predata 表示每个 batch 中的数据。
⑤使用 model 函数对每个 batch 的数据进行预测，得到 tensor 格式的结果。
⑥将 tensor 转化为 NumPy 数组，并将每个预测结果 pred 添加到 preds 列表中。
⑦将每个 batch 的真实标签添加到 label 列表中。
⑧对于 pred 数组中的每个元素，预测概率最大的标签值，并将该标签值添加到 preds 列表中。
⑨将 preds 列表中的元素压缩成一个一维数组，并将其转化为列表格式。此时，preds 列表中的每个元素即为预测标签。
⑩将 label 列表展开为一个一维列表，以得到整个测试集上的真实标签。
最终，可以对比 60 组测试集数据的预测标签 preds 和真实标签 label，以评估模型在测试集上的性能。
流程图如图 3-3 所示。

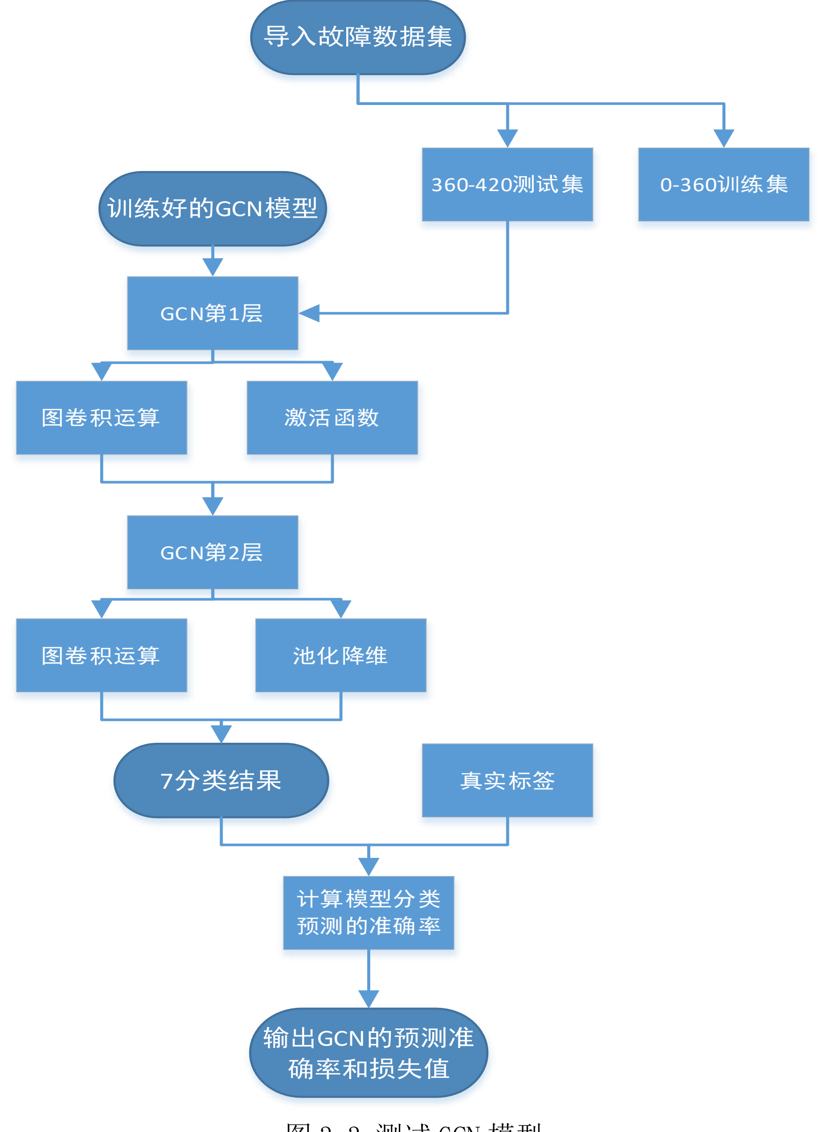


图 3-3 测试 GCN 模型

（3）结果分析

将测试集输入训练好的 GCN 模型，得到了 60 组测试集数据的分类预测结果，同时，用 CNN（卷积神经网络）对相同数据进行分类预测，以对比两种模型的预测准确率。

拟使用训练200次的Adam优化器2层GCN与训练200次/1000次的Adam/Adadelta优化器 2 层/3 层 CNN 进行对比，CNN 使用的数据集同样是 420 组故障数据。主要对比的内容是测试集得到的损失值和准确率大小，对比结果如表 3-1 所示。

表 3-1 GCN 与 CNN 的分类准确率和损失值

训练	Adadelta 优化器		Adam 优化器	
2 层 CNN 训练 200 次	Accuracy:0.615	Loss:1.518	Accuracy:0.929	Loss:0.052
3 层 CNN 训练 200 次	Accuracy:0.810	Loss:0.957		
2 层 CNN 训练 1000 次	Accuracy:0.731	Loss:0.904	Accuracy:0.964	Loss:0.037
3 层 CNN 训练 1000 次	Accuracy:0.746	Loss:0.899		
2 层 GCN 训练 200 次	Accuracy:0.967		Loss:0.009	

根据表格所示结果，在工业过程控制回路单故障诊断方面，使用训练 200 次的 2 层 Adam 优化器 GCN 在工业过程控制回路故障诊断准确率相较于训练 200 次和 1000 次的 2 层和 3 层 Adadelta 优化器 CNN 都明显更优，不管是准确率还是损失值都是 GCN 明显更优；而对于训练 200 次和 1000 次的 2 层 Adam 优化器 CNN，训练 200 次的 GCN 的准确率和损失值也要更高一筹，说明 GCN 需要的网络层数和训练次数都少于 CNN，且还能得到较好的分类结果。

综合上述结果可以判断：在工业过程简单控制回路的单故障诊断方面，使用带有先验知识约束，具有图结构的图卷积神经网络模型的性能明显优于普通的卷积神经网络。

2.2 控制回路多故障分类

（1）故障数据集

在工业过程控制回路的单故障诊断方面，可以看出 GCN 比 CNN 更优，但是由于是单故障的故障诊断，故障数据的复杂程度方面可能体现不出来图结构在卷积神经网络中的重要性，同时也考虑到现实中也可能出现控制回路中多个组件同时故障的情况，所以设计获取“阀门恒偏差+变送器恒偏差”、“阀门粘滞+变送器恒增益”、“控制器参数改变+阀门恒偏差”、“控制器噪声+阀门粘滞”四种单回路多故障的故障类型，用 GCN 进行多故障分类。

通过提高故障数据的复杂程度后，再分析 GCN 处理多故障分类的结果，以证明在 GCN 可以处理复杂情况下控制回路的故障分类问题。测试在工业过程的控制回路中使用有向图的构图方法，在处理更复杂的多故障数据时，是否仍然可以得到一个比较好的结果。

其中，“阀门恒偏差+变送器恒偏差”故障一共获取了 100 组，分别是 25 组阀门

负偏差+变送器正增益、25 组阀门负偏差+变送器负增益、25 组阀门正偏差+变送器正增益、25 组阀门正偏差+变送器负增益。

对于控制回路多故障一共获取了 280 组 150 个故障数据，同样提取其中 56s-95s 的 40 个数据构建多故障数据集。给上述 4 类故障打上 0、1、2、3 的标签，用于构建图标签数据。

将处理后的 280 组 4×40 的数据、图结构 4×4 的邻接矩阵和每类故障 1×4 的标签数据都转换成 Tensor 类型，以大小为 20 的 batch，打乱顺序后生成控制回路多故障的数据集。

此时得到的控制回路多故障数据集包含 14 个 batch，每个 batch 由标签为 x 的 80×40 节点特征集合、标签为 edge_index 的 2×80 边关系集合、标签为 y 的 20×7 的图分类标签集合组成。

（2）GCN 故障分类

多故障分类的 GCN 模型与单故障分类 GCN 一致，唯一区别是输入的数据集换成了多故障数据集，将前 250 组数据划分为训练集，后 30 组数据划分为训练集。

（3）结果分析

多故障分类的两层 GCN 模型经过了 200 次训练，其损失值在第 20 次训练时为 0.1105，而在 200 次训练后，损失值降至了 0.0017，模型训练的效果理想。故障分类的测试结果表现同样很好，对 30 组测试数据的真实标签预测准确率达到了 100%。这个结果足以证明 GCN 不仅能够在控制回路单故障分类中有很好的性能，在更加复杂的多故障分类中也可以取得优秀的结果。

此外，在构建单故障和多故障的图结构时，选择的是可以更好地表示变量之间的因果关系和流向性的有向图方法，而在单故障和多故障情况下，有向图都取得了很高的分类准确率。证明在基于 GCN 的工业过程控制回路故障分类问题上，使用有向图的效果很好。

4 结论

传统控制回路故障诊断方法存在成本高、时间长、效率低和精度不高等问题，难以应对复杂工业过程。卷积神经网络 (CNN) 作为深度学习的典型方法在数据结构上有一定局限性。本文介绍了图卷积神经网络 (GCN) 在深度学习中的原理和算法，并将其应用于控制回路的单一故障分类和多重故障分类问题。

在 GCN 的结构和原理介绍中，总结出以下几点：

1、GCN 相较于 CNN，通过图卷积操作可处理任意大小和任意连接的非欧几里德结构数据（如图形和社交网络），并能获得全局信息而无需多次卷积和池化。

2、在单一故障分类问题中，使用 GCN 模型相对于 CNN 模型，同样能够达到 95%以上的高分类准确率，且 GCN 的网络层数更少、训练次数更少，具有显著的效率优势。

在多重故障分类问题中，通过使用更复杂的多故障数据进行 GCN 分类模型的分析，可以证明，在工业控制回路中，选择变量间因果关系和流向性更强的有向图来构建图数据是明智的选择，基于有向图的 GCN 模型在单一和多重故障分类问题上表现出色。

不过，在实验中也发现一些问题，尽管 GCN 在控制回路图结构构建方面是创新的，但对于时序数据的预测并非 CNN 的强项。下一步研究方向可以考虑将动态图神经网络 (DGNN) 应用于控制回路故障诊断。与 GNN 不同，DGNN 具有动态图结构，可在训练中增加或减少节点和边，更适合处理控制回路故障数据这类具有动态特征的问题。

参考文献

[1] 林宗艺, 基于控制回路的故障诊断研究, 2006, 华北电力大学; 华北电力大学(保定).
[2] Chen Z, Xu J, Alippi C, et al. Graph neural network-based fault diagnosis: a review[J]. arXiv preprint arXiv:2111.08185, 2021.
[3] Li T, Zhou Z, Li S, et al. The emerging graph neural networks for intelligent fault diagnostics and prognostics: A guideline and a benchmark study[J]. Mechanical Systems and Signal Processing,
[4] 田烨. 基于二叉故障树的 SS4G 型电力机车控制回路故障诊断专家系统的研究[D].中国铁道科学研究院,2022.DOI:10.27369/d.cnki.gtdky.2022.000027.
[5] 苑博威. 基于特征挖掘与卷积神经网络的故障诊断方法研究[D].北京化工大学,2021.DOI:10.26939/d.cnki.gbhgu.2021.001249.
[6] 冯洁. 基于神经网络的控制系统故障诊断方法[D].北京化工大学,2012.
[7] 李金辉. 基于 ICA 的控制回路故障诊断方法[D].北京化工大学,2012.
[8] 臧海祥,郭镜玮,黄蔓云等.基于消息传递图神经网络的电力系统状态估计[J/OL].电网技术:1-16[2023-05-04]
[9] Monti F, Otness K, Bronstein M M. Motifnet: a motif-based graph convolutional network for directed graphs[C]//2018 IEEE Data Science Workshop (DSW). IEEE, 2018: 225-228.
[10] 李甜甜,张荣梅,张佳惠.图神经网络技术研究综述[J].河北省科学院学报,2022,39(02):1-13.DOI:10.16191/j.cnki.hbkx.2022.02.007.
[11] Wang Y, Qian S, Hu J, et al. Fake news detection via knowledge-driven multimodal graph convolutional networks[C]//Proceedings of the 2020 international conference on multimedia retrieval. 2020: 540-547.